

# State Identification for Planetary Rovers: Learning and Recognition

Olivier Aycard  
Leibniz-UJF  
46, Avenue Felix Viallet  
38 000 Grenoble, France  
Phone: (+33) 4.76.57.48.62  
Email: Olivier.Aycard@imag.fr

Richard Washington<sup>†</sup>  
Autonomy and Robotics Area  
NASA Ames Research Center  
MS 269-2, Moffett Field, CA 94035  
Phone: (+1) 650 604-1140  
Email: richw@ptolemy.arc.nasa.gov

## Abstract

*A planetary rover must be able to identify states where it should stop or change its plan. With limited and infrequent communication from ground, the rover must recognize states accurately. However, the sensor data is inherently noisy, so identifying the temporal patterns of data that correspond to interesting or important states becomes a complex problem. In this paper, we present an approach to state identification using second-order Hidden Markov Models. Models are trained automatically on a set of labeled training data; the rover uses those models to identify its state from the observed data. The approach is demonstrated on data from a planetary rover platform.*

## 1 Introduction

An autonomous mobile robot exploring or operating in an unknown environment needs to correctly identify fault states and environmental states in order to react to them appropriately. In the case of limited and delayed communication, such as for planetary rovers, human interaction is restricted, so these states can only be known through interpretation of the sensor information on board. Some states can only be identified by considering a temporal sequence of sensor information, not simply a snapshot. Additionally, the sensor data may be noisy, so simple descriptions of the sensor data (e.g., “current rising, steady, then falling”) may not directly correspond to the actual data.

Consider a planetary rover that traverses long distances over largely unknown terrain. If it ever rolls over a rock, it may want to turn its cameras and take a picture of the newly exposed rock surface abraded by the wheels. If it climbs a hill, it may be a good time to take images for localization and science site identifica-

tion. Predicting the exact moment when these states occur is difficult or impossible for long traverses.

An autonomous robot also needs to accurately identify its internal fault conditions to know whether it should stop its activity or continue. For example, a rock wedged in a rover’s drive train may be a recoverable error, but if left too long it may cause motor damage; on the other hand, frequent false alarms will decrease the overall effectiveness of the robot.

This paper describes a statistical approach to robot state identification, using Hidden Markov Models (HMMs). From a set of training data, the robot builds models that represent the statistical properties of the observations corresponding to each of a set of states. Then, at recognition time, the robot chooses the states whose models best approximate the observations.

The HMM approach is a flexible method for handling the large variability of complex temporal signals; for example, it is a standard method for speech recognition [7]. In contrast to dynamic time warping where heuristic training methods for estimating templates are used, stochastic modeling allows probabilistic and automatic training for estimating models.

The particular approach we use is the second-order HMM (HMM2). HMM2s have been shown to be efficient models to capture temporal variations in speech [5], in many cases surpassing first-order HMMs when the trajectory in the state space has to be accounted for. HMM2s have also been applied to mobile robot place recognition in indoor environments [1].

This paper is organized as follow. We first report related work. In section 3, we briefly present our mobile robot. In section 4, we define the HMM2 and describe the algorithms used for training and recognition. Section 5 is the description of our methodology. We discuss results in section 6 and give some conclusions and perspectives in section 7.

---

<sup>†</sup>NASA contractor with Caelum Research Corporation.



Figure 1: The Marsokhod rover.

## 2 Related Work

A variety of approaches to state estimation and fault diagnosis have been proposed in the control systems, artificial intelligence, and robotics literature.

Techniques for state estimation of continuous values, such as Kalman filters, can track multiple possible hypotheses [8, 10]. However, they must be given an *a priori* model of each possible state. One of the strengths of the approach presented in this paper is its ability to construct models from training data and then use them for state identification.

Qualitative model-based diagnosis techniques [2, 6] consider a snapshot of the system rather than its history. In addition, the system state is assumed to be consistent with a propositional description of one of a set of possible states. The presence of noisy data and temporal patterns negates these assumptions.

Hidden Markov Models have been applied to fault detection in continuous processes [9]; the model structure is supplied, with only the transition probabilities learned from data. In the approach in this paper, the HMM learns without prior knowledge of the models.

## 3 Marsokhod rover

The rover used in these experiments is a Marsokhod rover (see figure 1), a medium-sized planetary rover originally developed for the Russian Mars exploration program; in the NASA Marsokhod, the instruments and electronics have been changed from the original.

The rover has six wheels, independently driven<sup>1</sup>, with three chassis segments that articulate independently. It is configured with imaging cameras, a spectrometer, and an arm. The Marsokhod platform has been demonstrated at field tests from 1993–99 in Russia, Hawaii, and deserts of Arizona and California; the field tests were designed to study user interface issues, science instrument selection, and autonomy technologies.

The Marsokhod is controlled either through sequences or direct tele-operation. In either case the rover is sent discrete commands that describe motion in terms of translation and rotation rate and total time/distance. The Marsokhod is instrumented with sensors that measure body, arm, and pan/tilt geometry, wheel odometry and currents, and battery currents. The sensors that are used in this paper are roll (angle from vertical in direction perpendicular to travel), pitch (angle from vertical in direction of travel), and motor currents in each of the 6 wheels.

The experiments in this paper were performed in an outdoor “sandbox,” which is a gravel and sand area about 20m x 20m, with assorted rocks and some topography. This space is used to perform small-scale tests in a reasonable approximation of a planetary (Martian) environment. We distinguish between the small (less than approx. 15cm high) and large rocks (greater than approx. 15cm high). We also distinguish between the one large hill (approx. 1m high) and the three small hills (0.3-0.5m high).

## 4 Second-order Hidden Markov Model

In this section, we briefly present the HMM2 and the algorithms used for learning and recognition. A very complete tutorial on HMMs and their applications to speech can be found in [7].

### 4.1 Definition

In a HMM2, the underlying state sequence is a second-order Markov chain. Therefore, the probability of a transition between two states at time  $t$  depends on the states in which the process was at time  $t-1$  and  $t-2$ . A HMM2  $\lambda$  is specified by:

- a set of states,  $S$ ;
- a 3 dimensional matrix  $A$  over  $S \times S \times S$

$$a_{ijk} = \text{Prob}(q_t = s_k / q_{t-1} = s_j, q_{t-2} = s_i) \quad (1)$$

$$= \text{Prob}(q_t = s_k / q_{t-1} = s_j, q_{t-2} = s_i, q_{t-3} = \dots)$$

<sup>1</sup>For the experiments, the right rear wheel had a broken gear, so it rolled passively.

with  $\sum_{k=1}^N a_{ijk} = 1$  for all  $1 \leq i, j \leq N$ , where  $N$  is the number of states in the model and  $q_t$  is the actual state at time  $t$ ;

- a mixture of Gaussians associated with each  $s_i \in S$ :

$$b_i(O_t) = \sum_{m=1}^M c_{im} \mathcal{N}(O_t; \mu_{im}, \Sigma_{im}), \quad (2)$$

where  $\sum_{m=1}^M c_{im} = 1$  and  $O_t$  is the input vector (i.e., the observation<sup>2</sup>) at time  $t$ .

The probability of the state sequence  $Q = q_1, q_2, \dots, q_T$  is defined as

$$Prob(Q) = \pi_{q_1} a_{q_1 q_2} \prod_{t=3}^T a_{q_{t-2} q_{t-1} q_t} \quad (3)$$

where  $\pi_{q_1}$  is the probability of state  $s_{q_1}$  at time  $t = 1$  and  $a_{q_1 q_2}$  is the probability of the transition  $s_{q_1} \rightarrow s_{q_2}$  at time  $t = 2$ .

## 4.2 The Viterbi Algorithm

Recognition of a given sequence of observations is performed by the Viterbi algorithm [3], which determines the most likely state sequence given an observation sequence. The Viterbi algorithm uses dynamic programming to compute the best partial state sequence to time  $t$  for all states. The most likely state sequence  $q_1, \dots, q_T$  is obtained by keeping track of back pointers for each computation of which previous transition leads to the maximal partial path probability.

## 4.3 The Baum-Welch Algorithm

Model learning is performed using the maximum likelihood estimation criterion that determines the best model parameters according to the corpus of items. It must be noted that this criterion does not try to separate models like a neural network, but only tries to increase the probability that a model generates its corpus independently of what the other models can do. Intuitively, this algorithm counts the number of occurrences of each transition between the states in the training corpus. Each count is weighted by the probability of the alignment (state, observation).

## 5 Application to rover state identification

To allow the Marsokhod to learn to recognize particular situations while it is exploring the “sandbox”,

<sup>2</sup>An observation is defined as the measure of one or several sensors at a given time.

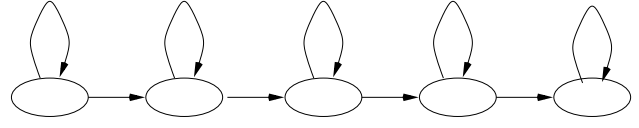


Figure 2: Topology of states used for each model of situation

we have to address several major issues: defining a set of situations; defining a second-order hidden Markov model associated with each situation; collecting a corpus of observations during several runs and labeling this corpus by finding temporal borders of each situation that the robot has observed during its run.

### 5.1 The set of situations

Currently, we model six distinct situations that are representative of a typical outdoor exploration environment: when the robot is climbing a small rock on its left (resp. right) side, a big rock on its left side<sup>3</sup>, a small (resp. big) hill, and a default situation of level ground. These situations are chosen for the fact that they are repeatable and human-observable (for labeling); internal fault identification would have required instrumenting the rover to cause faults on command, which is not currently possible on the Marsokhod.

We make the hypothesis that two or more of these situations cannot overlap (e.g., a small rock on the right side while climbing a big hill). This set of items is a complete description of what the mobile robot can see during its runs. All other unforeseen situations, like flat rocks or holes, are treated as noise.

### 5.2 The model to represent each situation

In the formalism described in section 4.1, each situation to be recognized is modeled by a HMM2 whose topology is depicted in figure 2. This topology is well suited for the type of recognition we want to perform. Recognition begins in the leftmost state, and each time an event characterizing the situation is recognized it advances to the right. When the rightmost state has been reached, the recognition of the situation is complete. The five-state model has been chosen experimentally to give the best rate of recognition.

### 5.3 Corpus collecting and labeling

We built six corpora to train a model for each situation. For this, our mobile robot made approximately

<sup>3</sup>The situation of a big rock on the right side was not considered because of the non-functional right-side wheel.

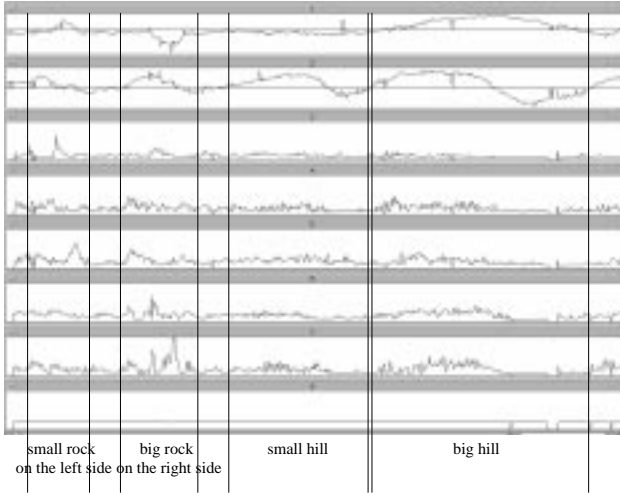


Figure 3: Segmentation and labeling of a run.

fifty runs in the sandbox. For each run, the robot received one discrete translation command ranging from three meters to twenty meters. Rotation motions are not part of the corpus. Each run contains different situations, but each run is unique (i.e., the area traversed and the sequence of situations during the run is different each time). A run contains not only one situation but all the situations seen while running. For each run, we noted the situations seen during the run, for later segmentation and labeling purposes.

As hidden Markov models have the ability to model signals whose properties change with time, we have to choose a set of sensors (as the observation) that have noticeable variations when the Marsokhod is crossing a rock or a hill. From the sensors described in section 3, we identified eight such sensors: roll, pitch, and the six wheel currents. We define coarse rules to identify each situation (for segmentation and labeling by humans):

- When the robot crosses a small (resp. big) rock on its left, we notice a distinct sensor pattern. In all cases, the roll sensor shows a small (resp. big) increase when climbing the rock, then a small (resp. big), sudden decrease when descending from the rock. These two variations usually appear sequentially on the front, middle, and rear left wheels. The pitch sensor always shows a small (resp. big) increase, then a small (resp. big), sudden decrease, and finally a small (resp. big) increase. There is little variation on the right wheels.
- When the robot crosses a small rock on its right side, we observe variations symmetric to the case of a small rock on the left side.

- When the robot crosses a small (resp. big) hill, the pitch sensor usually shows a small (resp. big) increase, then a small (resp. big) decrease, and finally a small (resp. big) increase. There is not always variation in the roll sensor. However, there is a gradual, small (resp. big) increase followed by a gradual, small (resp. big) decrease on all (or almost all) the six wheel current sensors.

These rules are used to segment and label each run. An example of segmentation and labeling is given in figure 3. The sensors are in the following order (from the top): roll, pitch, the three left wheel currents, and the three right wheel currents. A vertical line marks the beginning or the end of a situation. The default situation alternates with the other situations. The sequence of situations in the figure is the following (as labeled in the figure): small rock on the left side, default situation, big rock on the right side, default situation, small hill, default situation, and big hill.

#### 5.4 Model training

As we want to compare different possibilities, we perform a separate model training for each of three sets of input data. The observation used as input of each model to train consists of:

- eight coefficients: the first derivative (i.e., the variation) of the values of the eight sensors used for segmentation.
- six coefficients: the first derivative (i.e., the variation) of the values of the six wheel current sensors.
- two coefficients: the first derivative (i.e., the variation) of the values of the roll and the pitch sensors.

Each training uses segmented data, and each model is trained independently with its corpus.

The three different model trainings are performed for two reasons. First is to check whether the eight sensors used for the segmentation are necessary to learn and recognize situations, or whether a subset set is sufficient. Second, we want to be able to recognize situations even if one or more sensors do not work; e.g., if some wheel sensors do not work it will affect (during recognition) the models using the six wheel current sensors or the eight sensors but not the models using just the roll and the pitch sensor.

#### 5.5 The recognition phase

The robot's environment is described using a grammar that accepts only certain sequences of models. In this grammar, all the HMM2s are chained in a bigger HMM on which the Viterbi algorithm is used. The

best sequence of states determines the ordered list of situations seen during the run. The list of situations is known definitively when the run is completed.

The goal of recognition is to spot the five situations (small rock on the left or right; big rock on the left; small or big hill) while the robot moves in the sandbox. The default situation model connects two items much like silence between two words in speech recognition.

During the recognition phase, the robot was operated as for corpus collecting. We took approximately 40 acquisitions and used the six trained models to perform the recognition. A situation is recognized if it is detected by the corresponding model close to its real geometric position. A few types of errors can occur:

**Insertions:** the robot sees a non-existing situation.

**Deletions:** the robot misses a situation;

**Substitutions:** the robot confuses two situations.

We perform three independent recognitions (corresponding to the three learning situations).

## 6 Results and discussion

The results are presented first as confusion matrices, where an element  $c_{ij}$  is the number of times the model  $j$  has been recognized when the right answer was situation  $i$ , and second with the global rate of recognition, substitution, omission and insertion. In each confusion matrix, the acronyms used are: BL = big rock on the left, SL = small rock on the left, SR = small rock on the right, BH = big hill, and SH = small hill.

### 6.1 Experiment with eight sensors

	BL	SL	SR	BH	SH	Ins
BL	19	3	1	-	-	9
SL	3	25	-	-	-	12
SR	1	2	31	-	1	26
BH	1	-	-	20	2	15
SH	-	-	-	1	23	28
Omi	1	1	-	-	-	-
Total	25	31	32	21	26	90
Reco	76%	81%	97%	95%	88%	

Table 1: Confusion matrix of situations, eight sensors.

For eight sensors, as each situation can be easily distinguished from the others, the global rate of recognition is excellent (87%) (see tables 1, 2). Small (resp. big) rocks on the left are sometimes confused with big (resp. small) rocks on the left; the signal provided

	8 sensors		6 sensors		2 sensors	
	#	%	#	%	#	%
Seen	135	100	135	100	135	100
Recognized	118	87	113	84	82	61
Substituted	15	11	21	15	50	37
Omitted	2	2	1	1	3	2
Inserted	90	67	124	92	42	31

Table 2: Global rate of recognition, comparing 8 sensors, 6 sensors, and 2 sensors.

by the sensors does not contain the information for discriminating these two models. In fact, the variations on the sensors are nearly the same. The only criterion which distinguishes these two models is the amplitude of the variation on the three left wheels, and visibly it is not sufficient. The small rocks on the right are perfectly recognized. This situation has a very distinctive pattern, and only with difficulty can it be confused with another. The fact that we could not learn and recognize a situation where the robot is crossing a big rock on its right avoids any confusion.

The major problem is the high rate of insertion. This rate is due to the noise of the sensors being recognized as a situation. This is especially the case for situations characterized only by small variations on a part (or all) of the set of sensors, in particular the crossing of a small hill.

### 6.2 Experiment with six sensors

	BL	SL	SR	BH	SH	Ins
BL	17	5	1	-	-	10
SL	4	24	2	-	1	19
SR	3	-	29	-	1	44
BH	-	1	-	20	1	19
SH	1	-	-	1	23	32
Omi	-	1	-	-	-	-
Total	25	31	32	21	26	124
Reco	68%	77%	91%	95%	88%	

Table 3: Confusion matrix of situations, six sensors.

With six sensors, the global rate of recognition is still very good (see tables 3, 2). There is only four more percent of substitutions due to the loss of information used to distinguish situations. On the other hand, the rate of insertion increased by 25%. With only the six wheel current sensors, nearly one recognition out of two is an insertion. The six wheel current sensors are very noisy, and the roll and pitch sensors

are useful to distinguish between simple noise and real situations. This explains the increase of the insertions.

### 6.3 Experiment with two sensors

	BL	SL	SR	BH	SH	Ins
BL	15	4	-	6	1	1
SL	2	17	1	-	9	15
SR	2	1	27	1	5	8
BH	5	-	-	14	2	6
SH	-	7	4	-	9	12
Omi	1	2	-	-	-	-
Total	25	31	32	21	26	42
Reco	60%	55%	84%	67%	35%	

Table 4: Confusion matrix of situations, two sensors.

With only the roll and pitch sensors, the global rate of recognition remains good, and the rate of insertions significantly decreases (see tables 4, 2). In fact, these two sensors are not too noisy, and when there is a variation on these sensors it generally corresponds to a real situation. But these two sensors do not provide sufficient information to distinguish between situations, which is why there is a high rate of substitution.

## 7 Conclusion and perspectives

In this paper, we have presented a new method to learn and recognize situations in an outdoor environment with second-order hidden Markov models. One of the main interests of this work is the specification of an automatic learning algorithm of the environment that allows recognition of typical situations.

This method gives very good results, and has a good robustness to noise, verifying that HMM2s are well suited to learn and recognize situations during exploration of an outdoor environment. From the results of experiments, we can draw some conclusions. The best way to perform recognition is with eight sensors: the rate of recognition is a little bit better than for six sensors and the rate of insertion is very smaller. This can be explained by the fact that the six wheels current sensors are very noisy, and the use of the roll and pitch sensors, which are not too noisy, can distinguish between a situation to recognize and a simple noise on the current wheel sensors. Nonetheless, the learned models in the two last experiments can be useful in long exploration where sensors can fail.

Our method can be extended to fault detection, for example broken wheels or sensor failure. In fact, we

can build one model of a particular situation where all sensors work and several models of this situation where one or several sensors are broken: for example a model of a big rock on the right side and a model of a big rock on the right when the front left wheel is broken. Using these models, we can recognize situations associated with the state of the sensors of the robot.

To recognize situations online during exploration, we can use a variant of the Viterbi algorithm called Viterbi-block [4]. This algorithm is based on a local optimum comparison of the different probabilities computed by the Viterbi algorithm during time-warping of a shift-window of fixed length in the signal and the different HMMs. This algorithm can recognize situations a few meters after they have been seen.

## References

- [1] O. Aycard, P. Laroche, and F. Charpillet. Mobile robot localization in dynamic environment using place recognition. In *Proc. of ICRA '98*, 1998.
- [2] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artif. Intelligence*, 32:100–117, 1987.
- [3] G.D. Forney. The Viterbi Algorithm. *IEEE Transactions*, 61:268–278, March 1973.
- [4] A. Kriouile, J.-F. Mari, and J.-P. Haton. Some improvements in speech recognition algorithms based on HMM. In *Proceedings ICASSP'90*, 1990.
- [5] J.-F. Mari, J.-P. Haton, and A. Kriouile. Automatic word recognition based on second-order hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 5, January 1997.
- [6] N. Muscettola, P. P. Nayak, B. Pell, and B. C. Williams. Remote agent: To boldly go where no AI system has gone before. *Artificial Intelligence*, 103(1/2), August 1998.
- [7] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2), February 1989.
- [8] H. E. Rauch. Intelligent fault diagnosis and control reconfiguration. *IEEE Ctrl. Sys.*, 14(3), 1994.
- [9] P. Smyth. Hidden Markov models for fault detection in dynamic systems. *Pattern Recognition*, 27(1):149–164, January 1994.
- [10] A. S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12:601–611, 1976.